



# XML Fragment Interchange

W3C Candidate Recommendation 12 February 2001

This version:

<http://www.w3.org/TR/2001/CR-xml-fragment-20010212> (XML)

Latest version:

<http://www.w3.org/TR/xml-fragment>

Previous versions:

<http://www.w3.org/1999/06/WD-xml-fragment-19990630.html>

<http://www.w3.org/1999/06/WD-xml-fragment-19990630.xml>

<http://www.w3.org/TR/1999/WD-xml-fragment-19990412.html>

<http://www.w3.org/TR/1999/WD-xml-fragment-19990412.xml>

<http://www.w3.org/TR/1999/WD-xml-fragment-19990303.html>

<http://www.w3.org/TR/1999/WD-xml-fragment-19990303.xml>

Editors:

Paul Grosso, Arbortext <[pgrosso@arbortext.com](mailto:pgrosso@arbortext.com)>

Daniel Veillard, W3C <[veillard@w3.org](mailto:veillard@w3.org)>

Copyright © 2001 W3C<sup>®</sup> (MIT, INRIA, Keio), All Rights Reserved. W3C liability, trademark, document use, and software licensing rules apply.

---

## Abstract

The XML standard supports logical documents composed of possibly several entities. It may be desirable to view or edit one or more of the entities or parts of entities while having no interest, need, or ability to view or edit the entire document. The problem, then, is how to provide to a recipient of such a fragment the appropriate information about the context that fragment had in the larger document that is not available to the recipient. The XML Fragment WG is

chartered with defining a way to send fragments of an XML document—regardless of whether the fragments are predetermined entities or not—without having to send all of the containing document up to the part in question. This document defines Version 1.0 of the [eventual] W3C Recommendation that addresses this issue.

## Status of this Document

This specification is being put forth as a Candidate Recommendation by the XML Core Working Group. This document is a revision of the Working Draft dated 1999 June 30 which had incorporated suggestions received during last call review, comments, and further deliberations of the W3C XML Fragment Working Group. For background on this work, please see the XML Activity Statement. The Working Group believes this specification to be stable and invites implementation feedback during this period.

The duration of Candidate Recommendation is expected to last approximately three months (ending the end of April 2001). All persons are encouraged to review and implement this specification and return comments to the publicly archived mailing list [www-xml-fragment-comments@w3.org](mailto:www-xml-fragment-comments@w3.org).

Should this specification prove impossible to implement, the Working Group will return the document to Working Draft status and make necessary changes. Otherwise, the Working Group anticipates asking the W3C Director to advance this document to Proposed Recommendation.

This is still a draft document and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to cite a W3C Candidate Recommendation as other than "work in progress." A list of current W3C working drafts can be found at <http://www.w3.org/TR>.

## Table of Contents

- 1 Overview
- 2 Scope
- 3 Terminology
- 4 Fragment context information set

## 5 Fragment context specification notation

### 5.1 Overview of the fcs

### 5.2 Formal notation description

### 5.3 Semantics of a fragment context specification

### 5.4 An fcs example

#### 5.4.1 The parent Docbook book document

#### 5.4.2 The fragment body

#### 5.4.3 The fragment context specification document

## 6 Conformance

## Appendices

### A References

#### A.1 Normative References

#### A.2 Other References

### B Packaging and interchanging fragments (Non-Normative)

### C Examples (Non-Normative)

#### C.1 One element of a transaction record as a fragment

#### C.2 Use of external entities and MIME packaging

#### C.3 Indexes into a large document

### D Design Principles (Non-Normative)

### E Acknowledgments (Non-Normative)

### F Changes from Previous Public Working Drafts (Non-Normative)

#### F.1 Changes between the March 3 and April 2 WD

#### F.2 Changes between the April 2 and June 19 WD

#### F.3 Changes between the June 19 WD and the CR

---

## 1 Overview

The XML standard supports logical documents composed of possibly several entities. It may be desirable to view or edit one or more of the entities or parts of entities while having no interest, need, or ability to view or edit the entire document. The problem, then, is how to provide to a recipient of such a fragment the appropriate information about the context that fragment had in the larger document that is not available to the recipient.

In the case of many XML documents, it is suboptimal to have to receive and parse the entire document when only a fragment of it is desired. If the user asked to look at chapter 20, one shouldn't need to parse 19 whole chapters before getting to the part of interest. The goal of this activity is to define a way to enable processing of small parts of an XML document without having to process everything up to the part in question. This can be done regardless of whether the parts are entities or not, and the parts can either be viewed immediately or accumulated for later use, assembly, or other processing.

Conceptually, the holder of the complete source document considers a fragment of that document and, using the notation to be defined by this activity, constructs a **fragment context specification**. The object representing the fragment removed from its source document is called the **fragment body**. The fragment context specification and the fragment body are transmitted to the recipient. The storage object in which the fragment body is transmitted is called the **fragment entity**. (In some packaging schemes, the fragment context specification may also be embedded in the fragment entity.) The recipient processes the fragment context specification to determine the proper parser state for the context at the beginning of the fragment and uses that information to enable the XML parser to parse the fragment body. (The terms “sender,” “recipient,” “transmit,” are used throughout this document to describe the process of fragment interchange. It should be noted, however, that there are many feasible and useful scenarios for fragment interchange, and in some cases, the “sender” and “recipient” may be on the same machine, node, system, or network, and may even be the same tool in different guises.)

The challenge is that an isolated element from an XML document may not contain quite enough information to be parsed correctly. The goal of this activity is to enable senders to provide the remaining information required so that systems can interchange any XML elements they choose, from books or chapters all the way down to paragraphs, tables, footnotes, book titles, and so on, without having to manage each as a separate entity or having to risk incorrect parsing due to loss of context.

To accomplish these ends, this Recommendation defines:

- exact constraints on what portions of an XML document may constitute

- fragments to be supported by this Recommendation;
- the set of information (fragment context information) that allows for successful parsing as well as for viewing or editing of a fragment in a useful and important set of cases;
- the notation (i.e., language) in which this information will be described (the fragment context specification);
- some mechanisms for associating this information with a fragment.

## 2 Scope

This Recommendation enables interchanging portions of XML documents while retaining the ability to parse them correctly (that is, as they would be parsed in their originating document context), and, as far as practical, to be formatted, edited, and otherwise processed in useful ways.

The goal of this activity is to define a way to send fragments of an XML document—regardless of whether the fragments are predetermined entities or not—without having to send all of the containing document up to the part in question. The delivered parts can either be viewed or edited immediately or accumulated for later use, assembly, or other processing; what the receiving application does with the information—and issues involved with the possible “return” of such a fragment to the original sender—is beyond the scope of this activity. While implementations of this Recommendation may serve as part of a larger system that allows for “fragment reuse,” the many important issues about reuse of XML text and “concurrent multiple author environments” are beyond the scope of this Recommendation.

The point of the fragment context information is to provide information that is not available in the fragment body itself but that would be available from the complete XML document. Specifically, any information not available from the XML document (which may include an external subset) as a whole (plus knowledge of the location of the fragment body within the document) is out of scope for inclusion in the fragment context information. Such information may well be useful and important metadata in a variety of applications, but there are (or need to be) other mechanisms for handling this information.

This Recommendation considers fragments of XML as defined by [\[XML 1.0\]](#) and [\[XML Namespaces\]](#). It is explicitly noted that this version of this Recommendation does not take into account work such as that taking place in the XML Schema Working Group; insofar as such work by other currently active working groups places new requirements on a fragment interchange solution, those requirements would be input to a new version of the fragment interchange specification that may become a chartered activity at a later date.

It is also explicitly noted that this Recommendation does not consider interchange of information that is not well-formed XML; in particular, issues specific to the interchange of fragments of SGML (including HTML)—other than such SGML that is, in fact, also well-formed XML—are not within scope of this Recommendation.

### 3 Terminology

This list is sorted “logically” as opposed to alphabetically. In an entry, phrases in parentheses are “optional” modifiers; whether they are used explicitly or not, we are still talking about the same thing for the purposed of this Recommendation.

#### (well-formed) XML document

defined in [\[XML 1.0\]](#), [Well-formed XML documents](#)

#### (well-formed) (external) (parsed) entity

defined in [\[XML 1.0\]](#), [production \[78\] extParsedEnt](#)

#### (well-)balanced

A region (consecutive sequence of characters) of an XML document is said to be (well-)balanced if it matches [production \[43\] content of \[XML 1.0\]](#). Informally this means that, if the region includes any part of the markup of any construct, it contains all of the markup of that construct (e.g., in the case of elements, all of both the start and end tag).

#### fragment

A general term to refer to part of an XML document, plus possibly some extra information, that may be useful to use and interchange in the absence of the rest of the XML document. See the rest of the fragment-related terms when a more precise definition is required.

### **fragment interchange**

The process of receiving and/or parsing of a fragment by a fragment-aware application.

### **fragment body**

A well-balanced region of an XML document being considered as (logically and/or physically) separate from the rest of the document for the purposes of defining it as a fragment. Also, that part of a fragment entity that consists solely of the well-balanced region from the complete XML document. When it is important to indicate that a reference is specifically to the version of the fragment body still physically part of the originating (parent) document, this document will use the term “fragment body *in situ*.”

### **context information**

The abstract set of information—divorced from any particular language/syntax/notation—that constitutes the “parser state” at the point when a parser processing the complete XML document encounters (but has not yet processed) the first character of (what would be) the fragment body.

### **(fragment) context (information)**

(sometimes abbreviated fci) The subset of the context information that we decide will be expressible in any fragment context specification language. Also the abstract set of information represented by a particular fragment context specification.

### **fragment context specification**

(sometimes abbreviated fcs) A valid string in the language (notation) that this Recommendation defines that describes a set of fragment context

information. Also the particular string in a fragment entity or fragment package that describes the fragment's context information.

### **package [verb]**

To associate in some specified way a fragment body with a fragment context specification. This may include some way of combining both into a single XML-encoded object; combining both in some multipart MIME or archiving encoding; or linking the two via some sort of referencing, co-referencing, or third-party referencing scheme.

### **fragment entity**

The storage object in which the fragment body is stored and/or transmitted during the process of fragment interchange.

### **(fragment) package [noun]**

The object actually transmitted during the process of fragment interchange. Though one might expect this is the same thing as a fragment entity, the terms may or may not be synonyms in all cases; one could define a packaging mechanism whereby the fragment context specification is transmitted without the fragment body (which presumably gets retrieved later) in which case the fragment package is the fragment context specification, and the fragment entity gets retrieved later.

### **fragment context specification document**

As defined in this Recommendation, a valid fragment context specification (fcs) is a well-formed XML document. Therefore, when considered as a document, an fcs is sometimes referred to as a fragment context specification document (or fcs document). A fragment context specification document may also be a fragment package (i.e., it may be the actual object transmitted to effect fragment interchange).

### **send/receive (and sender/recipient)**

In the context of this Recommendation, words such as send/receive (and



sender/recipient) are used to describe the general process of fragment interchange. There are many feasible and useful scenarios for fragment interchange, and in some cases, the “sender” and “recipient” may be on the same machine, node, system, or network, and may even be the same tool in different guises. The only constant assumption is that the sender has access to and knowledge of the entire (parental) document from which the fragment comes, and the recipient is in possession only of the fragment package (though nothing in this Recommendation precludes the possibility of the recipient using the information in the fragment package, if available, to attempt to fetch more information from the sender).

## 4 Fragment context information set

In this section, numbers in brackets refer to productions in [\[XML 1.0\]](#). The following information shall constitute the complete fragment context information (fci) set:

1. A reference to the external subset (extSubset [30]), by specifying an ExternalID [75] for it.
2. Internal subset information using some or all of the following:
  1. A reference to an “externalized copy” of the internal subset (presumably generated by placing the internal declarations into a storage object such as extSubset [30]), presumably by specifying an ExternalID [75] for it.
  2. Some or all of markupdecl [29] and/or PEReference [69] allowed in an XML document's internal subset; note that PEReference implies expansion of what could be more external entities; also note that markupdecl includes comments, processing instructions, and declarations for elements, attribute lists, entities, and notations.
3. Ancestor information for the fragment body.
4. Sibling information for the fragment body.
5. Sibling information for any of the ancestors.
6. Element content (aka descendant) information for any of the ancestors or siblings.
7. Attribute information (attribute name and value) for:

1. any of the ancestors;
  2. any of the siblings of the fragment body;
  3. any of the siblings of any of the ancestors;
  4. any of the descendants of any of the ancestors or siblings.
8. A reference to the original/parental document by specifying an ExternalID [75] for it.
  9. A reference to the fragment body within the original/parental document by specifying an ExternalID [75].

From the above list, the following items affect proper (validating) parsing of the fragment:

- External subset
- Internal subset
- (Preceding) Siblings of the fragment body

The following items, while they cannot affect proper parsing, are usually considered part of the basic, structural XML parse tree:

- Ancestors
- (Preceding) Siblings of ancestors
- Following siblings of the fragment body and its ancestors
- Ancestor and sibling descendants
- Attributes

The following items, while not usually considered part of the basic, structural XML parse tree, are clearly definable pieces of information known or computable by any XML processor that is processing the parent document:

- XML declaration information of the parent document. Note that we have defined a fragment package to be an XML document. That is, the fragment package would contain its own XMLDecl-like information as necessary, so the fci itself need not include that information.
- A reference to the parent document.
- A reference to the fragment body *in situ*.

## 5 Fragment context specification notation

### 5.1 Overview of the fcs

The previous section defined the logical set of information possible in a fragment context. This section describes the notation in which to express a specific fragment context specification. All information would be optional; how much gets included in any particular fragment context specification is up to the sender and recipient, and how this gets determined is outside of the scope of this Recommendation.

**Note:**

While what gets included in any particular fragment context specification is outside of the scope of this Recommendation, some knowledge of the target application can help determine an appropriate level for the fcs. For example, if the target application is a user agent that will use Cascading Style Sheets (CSS) to display the fragment, the following information is necessary and sufficient given the current level of CSS selector capability: previous siblings of the fragment body, all ancestors of the fragment body, previous siblings of each of those ancestors, and all attributes on all those siblings and ancestors.

**Note:**

A given fragment context specification need not necessarily provide the ability to specify the complete set of fragment context information described in the previous section. In particular, because the XML 1.0 syntax for declarations is difficult to embed within an XML instance, the specific fragment context specification notation defined by this Recommendation does not allow for inline inclusion of internal subset information within the FCS. Internal subset information can only be included in the FCS via a reference to an “externalized copy” of the internal subset. Inline internal subset information may be more feasible once an instance syntax for declarations is defined, and such may be considered in future versions of the Fragment Interchange specification.

The syntax used is XML itself. In particular, a fragment context specification (fcs) is written as a single root XML element allowing up to five attributes and containing a subtree of other elements (possibly with attributes). The root element (and the element serving as the placeholder for the fragment body) comes from Fragment Interchange namespace, a specific namespace defined by this Recommendation; the contained subtree of elements comes from the namespace(s) of the document from which this fragment comes. For the purposes of exposition in this section, we assume namespace declarations such as the following are in force:

<pre>xmlns:f="http://www.w3.org/2001/02/xml-fragment" xmlns="http://www.oasis-open.org/docbook/DocbookSchema"</pre>
---

That is, within this example, *f* is the local prefix referring to the Fragment Interchange namespace defined by this Recommendation for fragment-interchange related components, and the default namespace is that in effect in the parent document at the beginning of the fragment body *in situ*.

The element type for the single root element for the fcs shall be *f:fcs* (where *f* is whatever namespace prefix is mapped to the Fragment Interchange namespace). It allows up to five attributes, each of whose value shall be a URI reference [[RFC 2396]]. The attribute names and the meaning of their values are as follows:

**extref**

a URI reference to the external subset

**intref**

a URI reference to the internal subset

**parentref**

a URI reference to the parent document

**sourcelocn**

a URI reference to the fragment body *in situ* within the parent document

The content of the `f:fcs` element shall be a subtree of elements (possibly with attribute value assignments) from the parent document's namespace. This subtree shall provide all the structural context for the fragment body including various information about ancestor and sibling elements and attributes by mimicking the (relevant) context within this parent document. No data characters (mixed content) are allowed within the `f:fcs` element. The special empty element `f:fragbody` shall be used to indicate the placement of the fragment body within the specified context. It has one significant attribute with meaning as follows:

### fragbodyref

a URI reference [\[RFC 2396\]](#) to the fragment body

For example, consider a fragment body that consists of `listitems` 2 and 3 of an `orderedlist` (indicated to be enumerated with arabic numbers by the `numeration` attribute on the `orderedlist` element) within the second `sect1` within the first chapter within the first part within the body of a book. Assume that the external subset (aka "DTD") is in the file `Docbook.dtd` on the OASIS Open web server, the parent document is in `mybook.xml` on Acme's web server, and that there need be no internal subset given as part of the `fcs`. Then the `fcs` for this fragment body might look like:

```
<f:fcs xmlns:f="http://www.w3.org/2001/02/xml-fragment"
      extref="http://www.oasis-open.org/docbook/docbook/3.0/docbook.dtd"
      parentref="http://www.acme.com/~me/mydocs/mybook.xml"
      xmlns="http://www.oasis-open.org/docbook/DocbookSchema">
  <book>
    <part>
      <chapter>
        <sect1/>
        <sect1>
          <orderedlist numeration="arabic">
            <listitem/>
            <f:fragbody/>
          </orderedlist>
        </sect1>
      </chapter>
    </part>
  </book>
</f:fcs>
```

## 5.2 Formal notation description

A formal notation for the `fcs` element used in the examples of the previous section follows. Therein, the following terms are defined in either the "Extensible

Markup Language (XML) 1.0" ([XML 1.0]) or "Namespaces in XML" ([XML Namespaces]) Recommendations: NCName, AttValue, Eq, S, Attribute, STag, ETag, EmptyElemTag, CharData, Reference, CDsect, PI, Comment, prolog, and Misc.

### *Fragment Context Specification Element*

[1]	FCSelement	::=	<u>FCSstag</u> <u>S?</u> <u>FCSelementContent</u> <u>S?</u> <u>FCSetag</u>	
[2]	FCSstag	::=	'<' <u>NCName</u> ':'fcs' (( <u>S</u> 'extref' <u>Eq</u> <u>AttValue</u> )   ( <u>S</u> 'intref' <u>Eq</u> <u>AttValue</u> )   ( <u>S</u> 'parentref' <u>Eq</u> <u>AttValue</u> )   ( <u>S</u> 'sourcelocn' <u>Eq</u> <u>AttValue</u> )   ( <u>S</u> <u>Attribute</u> ))* <u>S?</u> '>'	[Constraint: FCS Constraint: Fragment Namespace]
[3]	FCSelementContent	::=	<u>EmptyElemTag</u>   <u>STag</u> <u>FCScontent</u> <u>ETag</u>   <u>FCSfragbody</u>	[Constraint: FCS Constraint: Exactly One Fragbody]
[4]	FCSfragbody	::=	'<' <u>NCName</u> ':'fragbody' (( <u>S</u> 'fragbodyref' <u>Eq</u> <u>AttValue</u> )   ( <u>S</u> <u>Attribute</u> ))* <u>S?</u> '/>'	[Constraint: FCS Constraint: Same Namespace Prefix]
[5]	FCSetag	::=	'</' <u>NCName</u> ':'fcs' <u>S?</u> '>'	
[6]	FCScontent	::=	( <u>FCSelementContent</u>   <u>CharData</u>   <u>Reference</u>   <u>CDsect</u>   <u>PI</u>   <u>Comment</u> )*	

#### **Constraint: FCS Constraint: Fragment Namespace**

The namespace prefix represented by NCName in the production for FCSstag (and, therefore necessarily, FCSetag) must have been declared on one of the ancestors of the FCS element and must be associated with the Fragment Interchange Namespace URI defined in this Recommendation.

#### **Constraint: FCS Constraint: Exactly One Fragbody**

There must be exactly one fragbody (FCSfragbody) element in the fcs.

#### **Constraint: FCS Constraint: Same Namespace Prefix**

The namespace prefix (NCName) used in the production for FCSfragbody must be the same as that used in the production for FCSstag.

[Definition: The **fragment Interchange namespace** shall be associated with the following URI: <http://www.w3.org/2001/02/xml-fragment>.]

In the productions for FCSstag and FCSfragbody, there can be any number of other attribute assignments, all of which are ignored by the fragment context specification processor. Per XML 1.0 compliance, there can be at most one assignment to any given attribute including the specifically mentioned attributes. (Since there is no “and” connector in EBNF, this restriction is difficult to show directly in the EBNF, hence this restriction in prose; however, this prose restriction is normative.)

In the production for FCScontent, the fragment processor can optionally expand any References that it can expand. Then all CDSections, PIs, Comments, remaining References, and CharData (including whitespace, S) are ignored by the FCS processor.

**Note:**

If a Reference in FCScontent is expanded and the expansion includes element structure, that element structure is considered part of the fcs as it would if it had been included originally in its expanded form in the fcs. However, since expansion of any Reference in FCScontent is optional on the part of the fragment context specification processor, any sender for which such expansion is important should do the expansion when creating the fragment package.

*Fragment Context Specification*

[7] FCS ::= prolog FCSelement [Constraint: FCS Constraint: Well-formed, namespace complete]  
Misc\*

**Constraint: FCS Constraint: Well-formed, namespace complete**

A fragment context specification shall constitute a well-formed document conforming to the “Extensible Markup Language (XML) 1.0” ([XML 1.0]) and

“Namespaces in XML” ([XML Namespaces]) Recommendations. In particular, if there are entity references in the fcs, the fcs document must comply with the Entity declared well-formedness constraint per the “Extensible Markup Language (XML) 1.0” ([XML 1.0]) Recommendation. (Appropriate declarations would appear in the internal subset of the fcs document.) Furthermore, for any use of namespaces, the fcs document must comply with the Namespace declared namespace constraint per the “Namespaces in XML” ([XML Namespaces]) Recommendation.

**Note:**

Generally, a fragment context specification document would be the well-formed document consisting simply of the `f:fcs` element (and its contents) with no prolog. However, a prolog is always allowable and might be necessary when some declarations are required to satisfy the Entity declared well-formedness constraint.

**Note:**

Since all of the components in prolog are optional, an FCSelement by itself is an allowable fragment context specification, and this Recommendation does not preclude some packaging scheme from combining an FCSelement along with a fragment body as shown in some of the examples in B Packaging and interchanging fragments and C Examples.

## 5.3 Semantics of a fragment context specification

The previous section formally defines a fragment context specification to be a well-formed XML document consisting of a single `f:fcs` element with optional attributes and some content. The `f:fcs` element's content consists of optional stuff from the parent document (from which the fragment body is taken) plus a single `f:fragbody` element with optional attributes. The `f:fcs` and `f:fragbody` elements come from a namespace defined by this Recommendation and have certain specific semantics relative to fragment interchange as defined by this section.

While it is important to be able to package a fragment body with its fcs, it is



expected that a general XML-friendly packaging mechanism will be developed by the W3C that would satisfy this requirement. Meanwhile, this Recommendation defines a simple association mechanism that doesn't rely on a packaging scheme. Applications and interchange partners may agree on any packaging mechanism to aid in fragment interchange—this is beyond the scope of this Recommendation.

The `fcs` document is a well-formed XML document that (1) provides the fragment context and (2) provides a reference to the fragment body. Because it is well-formed, existing XML processors can be used to process `fcs` documents. To support this fragment interchange Recommendation, an application must also understand the semantics of the `f:fcs` and `f:fragbody` elements and their attributes and process accordingly.

Specifically, the `fragbodyref` attribute on the `fragbody` element is a URI reference [\[\[RFC 2396\]\]](#) to the fragment body. A fragment-aware processor is expected to resolve this reference and process the referenced fragment body in the context specified by the `fcs`. None of the attributes on the `fcs` element have required semantics with respect to fragment processing; they are provided (optionally) for the application's use at its discretion.

**Note:**

For example, a browser might bring up an `fcs` document, “expand” the reference to the fragment body (i.e., put a copy of the fragment body in place of the `fragbody` element), and then ignore (e.g., not display) the part of the document that was originally the `fcs`, thereby displaying (in the proper context) only the part of the document that was originally the fragment body.

**Note:**

The `fragbody` element and its `fragbodyref` attribute are in many ways logically equivalent to an external entity reference or an XLink reference with an “embed” semantic.

## 5.4 An `fcs` example

The following example shows the complete set of information relative to interchanging the two `listitem`s for the Docbook book mentioned in [5.1 Overview of the fcs](#).

The parent document, in `~me/mydocs/mybook.xml` on Acme's web server, is a Docbook book document whose contents is outlined in the first subsection below. The fragment body of interest consists of `listitem`s 2 and 3 of the `orderedlist` (indicated to be enumerated with arabic numbers by the `numeration` attribute on the `orderedlist` element) within the second `sect1` within the first chapter within the first part within the body of this book. The external subset (aka "DTD") is in the file `docbook.dtd` on the OASIS Open web server.

#### 5.4.1 The parent Docbook book document

The following represents the parent document from which the fragment body in question comes.

```
<?xml version="1.0"?>
<!DOCTYPE book SYSTEM "http://www.oasis-open.org/docbook/docbook/3.0/docbook.
<book xmlns="http://www.oasis-open.org/docbook/DocbookSchema">
  <part>
    <chapter><title>The title for chapter one</title>
      <sect1><title>The title for section one in chapter one</title>
        <p>The first paragraph....</p>
        <p>....</p>
      </sect1>
      <sect1><title>The title for section two in chapter one</title>
        <p>An introductory paragraph preceding an ordered list.</p>
        <orderedlist numeration="arabic">
          <listitem><para>This is the first listitem in this ordered
            list.</para></listitem>
          <listitem><para>This is the second listitem within the
            second sect1 of the first chapter within the first part
            of a Docbook <quote>book</quote> document.</para></listitem>
          <listitem><para>And this is the next listitem.</para></listitem>
          <listitem><para>This is the fourth and last listitem.</para></list
        </orderedlist>
        <p>Another paragraph....</p>
      </sect1>
    </chapter>
    <chapter><title>More content</title>
      <p>More chapters, sections, paragraphs, and such....</p>
    </chapter>
  </part>
</book>
```

Note that the declaration of the default namespace on the `<book>` tag isn't required for fragment interchange, but is shown for the purposes of completeness of this example.

### 5.4.2 The fragment body

The following shows the fragment body in a separate file ready for interchange. For the purposes of this example, we are assuming that this is in the file `~me/mydocs/myfrag.xml` on Acme's web server.

```
<listitem><para>This is the second listitem within the
second sect1 of the first chapter within the first part
of a Docbook <quote>book</quote> document.</para></listitem>
<listitem><para>And this is the next listitem.</para></listitem>
```

### 5.4.3 The fragment context specification document

The following shows what the fcs document might look like for the above parent document and fragment body. If this were in the file (e.g., `myfrag.fcs`), when this file is sent to any recipient with a fragment-aware tool, that tool should be able to access and process the desired fragment body.

```
<f:fcs xmlns:f="http://www.w3.org/2001/02/xml-fragment"
  extref="http://www.oasis-open.org/docbook/docbook/3.0/docbook.dtd"
  parentref="http://www.acme.com/~me/mydocs/mybook.xml"
  xmlns="http://www.oasis-open.org/docbook/DocbookSchema">
  <book>
    <part>
      <chapter>
        <sect1/>
        <sect1>
          <orderedlist numeration="arabic">
            <listitem/>
            <f:fragbody fragbodyref="http://www.acme.com/~me/mydocs/myfrag
          </orderedlist>
        </sect1>
      </chapter>
    </part>
  </book>
</f:fcs>
```

Note that the `fragbodyref` value, which is a URI reference [\[\[RFC 2396\]\]](#), could be a URL, a file name, a MIME content id, etc., depending on the MIME type of the referenced resource. Also note that the `parentref` value above is only there for the information of the receiving application, but is not necessary for this example's operation. Likewise, the `extref` would only be necessary if the receiving application wanted to be able to do validation.

## 6 Conformance

A fragment conforms to this XML Fragment Interchange Recommendation if it adheres to all syntactic requirements defined in this Recommendation. A fragment is syntactically correct if all of the requirements specified in Section 5.2 are met.

Application software acting as recipient conforms to the XML Fragment Interchange Recommendation if it interprets all conforming XML fragments (as defined above) according to all required semantics prescribed by this Recommendation, and, for any optional semantics it chooses to support, supports them in the way prescribed. Specifically, conforming application software must be able to parse all conforming valid fragment context specification information whether it chooses to support its semantics or not. Application software acting as sender conforms to the XML Fragment Interchange Recommendation if it creates conforming XML fragments (as defined above) and, if including fragment context information, includes conforming fragment context information according to the requirements in section 4.

If fragment context information is included with a transmitted fragment, then it should conform according to the requirements in section 4.

## A References

### A.1 Normative References

#### RFC 2396

IETF RFC 2396: *Uniform Resource Identifiers (URI): Generic Syntax*. See <ftp://ftp.ietf.org/rfc/rfc2396.txt>

#### XML 1.0

World Wide Web Consortium. *Extensible Markup Language (XML) 1.0*. W3C Recommendation. See <http://www.w3.org/TR/REC-xml>

#### XML Namespaces

World Wide Web Consortium. *Namespaces in XML* W3C Proposed Recommendation. See <http://www.w3.org/TR/PR-xml-names>

#### Associating stylesheets

World Wide Web Consortium. *Associating stylesheets with XML documents* W3C Working Draft. See <http://www.w3.org/TR/WD-xml-stylesheet>

## A.2 Other References

### TR9601

OASIS (formerly SGML Open) *Fragment Interchange — SGML Open Technical Resolution 9601:1996*. OASIS (SGML Open) Technical Resolution. See <http://www.oasis-open.org/html/tr9601.html> for an online version

### MIME

IETF RFC 2045: *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*. See <http://www.imc.org/rfc2045>

### RFC 2387

IETF RFC 2387: *The MIME Multipart/Related Content-type*. See <http://www.imc.org/rfc2387>

### RFC 2392

IETF RFC 2392: *Content-ID and Message-ID Uniform Resource Locators*. See <ftp://ftp.ietf.org/rfc/rfc2392.txt>

### XML Fragment Requirements Document

World Wide Web Consortium. *XML Fragment Interchange Requirements W3C Note*. See <http://www.w3.org/TR/NOTE-XML-FRAG-REQ>

### XPointer WD

World Wide Web Consortium. *XML Pointer Language (XPointer) W3C Working Draft*. See <http://www.w3.org/TR/WD-xptr>

## B Packaging and interchanging fragments (Non-Normative)

It is a design goal of this Recommendation to define a fragment context specification to be a well-formed XML document. However, a fragment body itself need not be a well-formed document, but only well-balanced. While it is important to be able to package a fragment body with its fcs, it is expected that a general XML-friendly packaging mechanism—beyond the scope of this Recommendation—will be developed by the W3C that would satisfy this requirement. Meanwhile, applications and interchange partners may agree on any packaging mechanism to aid in fragment interchange. This appendix gives some non-normative examples of such possible packaging mechanisms.

The fcs element could be packaged along with the fragment body by combining

them into a single well-formed XML document. For the purposes of fragment interchange packaging, one could define a simple “document type” consisting of a “head” part containing the fcs (and, potentially, other) metadata followed by a “body” part containing the fragment body itself.

In the following template, *p* is defined as the local prefix referring to the namespace defined for the packaging structure, and *f*, as in previous sections, is the local prefix referring to the namespace defined by this Recommendation for fragment-interchange related components. (Note that this template example assumes that no explicit namespace prefixes are present in the fragment body. If the fragment body contains explicit namespace prefixes whose declarations are not also included in the fragment body, then additional namespace declarations would be necessary on the `<p:package>` or `<f:fcs>` element. If the parent document does not use namespaces at all, then no default namespace declaration is needed for the fcs or its package.)

The format of a complete fragment package might be outlined as follows:

```
<p:package xmlns:p="http://www.w3.org/2001/02/xml-package"
          xmlns:f="http://www.w3.org/2001/02/xml-fragment"
          xmlns="{the default namespace in effect at the start
                  of the fragment body in the parent document}">
  <f:fcs {the ref attributes on the fcs tag}>
    {the content of the fcs with no namespace prefixes
     necessary except that on the <f:fragbody/> element}
  </f:fcs>

  <p:body>
    {the fragment body with no namespace prefixes necessary}
  </p:body>
</p:package>
```

**Note:**

The above template includes indentation and blank lines to help display the overall structure of the package. However, all whitespace within the `p:body` element *is* significant and is therefore part of the fragment body. Therefore, the packaging process can introduce no whitespace (including record ends immediately following `<p:body>` and immediately preceding `</p:body>`) within the `p:body` element.

## C Examples (Non-Normative)

The following examples are designed in general to address the potential reference scenarios described in [XML Fragment Requirements Document].

## C.1 One element of a transaction record as a fragment

The user has an XML document that represents a customer's set of purchases at a bookstore, and the part of that document that represents the purchase of a particular book needs to be represented as a fragment.

Here is the original XML document for the transaction:

```
<?xml version="1.0"?>
<transaction TID="19990207-1234">
  <purchase>
    <book>
      <Author>Frank Herbert</Author>
      <Title>Dune</Title>
      <Edition>Hardcover Reissue edition (April 1984)</Edition>
      <ISBN>0399128964</ISBN>
      <Price currency="USD">18.87</Price>
      <Quantity>1</Quantity>
    </book>
    <book>
      <Author>J. R. R. Tolkien</Author>
      <Title>The Book of Lost Tales (The History of Middle-Earth)</Title>
      <Edition>Mass Market Paperback Reprint edition (June 1992)</Edition>
      <ISBN>0345375211</ISBN>
      <Price currency="USD">4.79</Price>
      <Quantity>1</Quantity>
    </book>
  </purchase>
  <refund RID="19990115-2">
    <reason TID="19981220-3214">Late delivery</reason>
    <value currency="USD">5.00</value>
  </refund>
  <payment>
    <client CID="123421"/>
    <value currency="USD">18.66</value>
    <creditcard type="MasterCard">
      <bank>BankBoston</bank>
      <owner>Joe J. Bill</owner>
      <serial>1234567890</serial>
      <expires>5/99</expires>
    </creditcard>
    <status>Waiting for approval</status>
  </payment>
</transaction>
```

Here is a fragment representing the second book element from the above document (the `sourceLocn` attribute on the `f:fcs` element is optional and is shown merely as an example):

```
<?xml version="1.0"?>
```

```
<p:package xmlns:p="http://www.w3.org/2001/02/xml-package">
  <p:fcs xmlns:f="http://www.w3.org/2001/02/xml-fragment"
    sourcelocn="http://acme.com/trans1234#root().child(1,purchase).child(1,transaction)">
    <transaction>
      <purchase>
        <book/>
      </purchase>
    </transaction>
  </p:fcs>

  <p:body>
    <book>
      <Author>J. R. R. Tolkien</Author>
      <Title>The Book of Lost Tales (The History of Middle-Earth)</Title>
      <Edition>Mass Market Paperback Reprint edition (June 1992)</Edition>
      <ISBN>0345375211</ISBN>
      <Price currency="USD">4.79</Price>
      <Quantity>1</Quantity>
    </book>
  </p:body>
</p:package>
```

## C.2 Use of external entities and MIME packaging

A user has an XML document that includes several external entities, and she wants to be able to interchange a fragment that includes a reference to the entities using MIME [\[\[MIME\]\]](#) packaging. (For references, see also [\[RFC 2387\]](#) and [\[RFC 2392\]](#).)

Here is the original document:

```
<?xml version="1.0"?>
<!DOCTYPE book SYSTEM "http://www.oasis-open.org/docbook/docbook/3.0/docbook.
<!ENTITY title "My Book">
<!ENTITY author "me">
<!ENTITY try SYSTEM "try.cgm" NDATA CGM-BINARY>
]>
<book>
  <part>
    <title>&title;</title>
    <introduction>This is my book ...</introduction>
    <author>&author;</author>
    <chapter type="intro">
      <sect1>The introduction ...</sect1>
    </chapter>
    <chapter>...</chapter>
    <chapter>
      <p>This is a paragraph within the third chapter within
the first part of a Docbook <quote>book</quote> document.</p>
      <p>And this is a succeeding paragraph.</p>
      <p>And an internal text entity reference &author;.</p>
      <p>And a reference to an unparsed entity (a CGM graphic):
        <graphic entityref="try"></graphic></p>
    </chapter>
    <chapter>...</chapter>
  </part>
```



```
</book>
```

Note that the DocBook DTD includes the following (which is therefore not included in the internal subset of this document):

```
<!NOTATION CGM-BINARY PUBLIC "ISO 8632/3//NOTATION Binary Encoding//EN">
```

Here is a fragment that represents the contents of the third chapter:

```
<?xml version="1.0"?>
<f:fcs xmlns:f="http://www.w3.org/2001/02/xml-fragment"
  xmlns="http://www.oasis-open.org/docbook/docbook/3.0/docbook.dtd"
  extref="http://www.oasis-open.org/docbook/docbook/3.0/docbook.dtd"
  intref="mybook.decls">
  <book>
    <part>
      <chapter type="intro"/>
      <chapter/>
      <chapter>
        <f:fragbody fragbodyref="chapter3.xml"/>
      </chapter>
    </part>
  </book>
</f:fcs>
```

Here is the corresponding fragment body:

```
<p>This is a paragraph within the third chapter within
the first part of a Docbook <quote>book</quote> document.</p>
<p>And this is a succeeding paragraph.</p>
<p>And an internal text entity reference &author;.</p>
<p>And a reference to an unparsed entity (a CGM graphic):
  <graphic entityref="try"></graphic></p>
```

Here is the associated internal subset:

```
<!ENTITY title "My Book">
<!ENTITY author "me">
<!ENTITY try SYSTEM "try.cgm" NDATA CGM-BINARY>
```

Here is the external entity (represented in Base 64 encoding, since this is really a binary entity):

```
ACEAABAiAAEQXWBEQyJTb3VyY2U6IEhTSSAvV01GLXRvLUNHTSBmaWx0ZXIq
LyBwZXJzaW9uIDEuMzU6IEhTSSAvV01GLXRvLUNHTSBmaWx0ZXIq
AAAQpgAAAAkAFxFGAAAA///EYQWigAQEYogyAAAAAB//3//AAARvWC3C1RJ
TUVTX1JPTUFODFRJTUVTX01UQUxJQWpUSU1FU19CT0xEeVRJTUVTX0JPTeRf
SVRBTElDCUhFTFZFVElDQRfIRUXWRVRJQ0Fft0JMSVFVRQ5IRUXWRVRJQ0Ff
```

```
Qk9MRBZIRUXWRVRJQ0FfQk9MRF9PQkxJUUVFB0NPVVJJRVIOQ09VUKlFUl9J
VEFMSUMMQ09VUKlFUl9CT0xEE0NPVVJJRVJfQk9MRF9JVEFMSUMGU1lNQk9M
ABHOAAABQgABAUEABAMqLTOR4gABAGEAACAmAAE9NJ9IIEIAASBiAAAgggAA
IKIAACDI95D0wAhqCzoAAACAQWj5cAa5/TEJikGGAogCUQGUGIACEAo+dD/
+v7g+TprYgACUKwAAQAEAAAAAAAAAABRgBxUggAAABKAGQAAFKCAAJAKAEg/
MoAAQlTb2l1IFRleHQAoABA
```

And here is an example of MIME packaging used to transmit the fragment context specification, the fragment body, the internal subset, and the external entity within a single stream such as a mail message:

```
Content-Type: multipart/related; boundary="/04w6evG8XlLl3ft";type="text/xml"

--/04w6evG8XlLl3ft
Content-Type: text/xml; charset=us-ascii
Content-ID: <part1>
Content-Disposition: attachment; filename="mybook.decls"

<!ENTITY title "My Book">
<!ENTITY author "me">
<!ENTITY try SYSTEM "cid:part2" NDATA CGM-BINARY>

--/04w6evG8XlLl3ft
Content-Type: image/cgm
Content-ID: <part2>
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="try.cgm"

ACEAABAiAAEQXwBEQyJTb3VyY2U6IEhTSSAvV01GLXRvLUNHTSBmawx0ZXIq
LyBWZXJzaW9uIDEuMzUgIiAiRGF0ZTogMTk5OS0wMS0xNyIRZgAB//8AARBi
AAAPgAAAAkAFxFGAAAA///EYQWIGAQEYogyAAAAAB//3//AAARVwC3C1Rj
TUVtX1JPTUFODFRJTUVtX0lUQUxJQWpUSU1FUl9CT0xEEVRJTUVtX0JPTeRf
SVRBTElDCUhfTFZFVElDQRFIrUXWRVRJQ0FfT0JMSVFVRQ5IRUXWRVRJQ0Ff
Qk9MRBZIRUXWRVRJQ0FfQk9MRF9PQkxJUUVFB0NPVVJJRVIOQ09VUKlFUl9J
VEFMSUMMQ09VUKlFUl9CT0xEE0NPVVJJRVJfQk9MRF9JVEFMSUMGU1lNQk9M
ABHOAAABQgABAUEABAMqLTOR4gABAGEAACAmAAE9NJ9IIEIAASBiAAAgggAA
IKIAACDI95D0wAhqCzoAAACAQWj5cAa5/TEJikGGAogCUQGUGIACEAo+dD/
+v7g+TprYgACUKwAAQAEAAAAAAAAAABRgBxUggAAABKAGQAAFKCAAJAKAEg/
MoAAQlTb2l1IFRleHQAoABA

--/04w6evG8XlLl3ft
Content-Type: text/xml; charset=us-ascii
Content-ID: <part3>
Content-Disposition: attachment; filename="chapter3.xml"

    <p>This is a paragraph within the third chapter within
the first part of a Docbook <quote>book</quote> document.</p>
    <p>And this is a succeeding paragraph.</p>
    <p>And an internal text entity reference &author;.</p>
    <p>And a reference to an unparsed entity (a CGM graphic):
      <graphic entityref="try"></graphic></p>

--/04w6evG8XlLl3ft
Content-Type: text/xml; charset=us-ascii

<?xml version="1.0"?>
<f:fcs xmlns:f="http://www.w3.org/2001/02/xml-fragment"
  xmlns="http://www.oasis-open.org/docbook/docbook/3.0/docbook.dtd"
  extref="http://www.oasis-open.org/docbook/docbook/3.0/docbook.dtd"
  intref="cid:part1">
  <book>
```

```
<part>
  <chapter type="intro"/>
  <chapter/>
  <chapter>
    <f:fragbody fragbodyref="cid:part3"/>
  </chapter>
</part>
</book>
</f:fcs>

--/04w6evG8X1L13ft--
```

### C.3 Indexes into a large document

The user has very large XML documents, possibly a gigabyte or more in size, and wishes to be able to view portions of the document without parsing the whole document. In order to do this the user creates an “index” for each document portion (fragment) that they wish to so address. The “index” consists of a fragment context specification in combination with a packaging mechanism designed for quick access to the fragment body. This should be used to view and browse documents with a flat structure, like HTML, on devices where only a part of the document can be parsed or rendered.

```
<?xml version="1.0"?>
<f:fcs xmlns:f="http://www.w3.org/2001/02/xml-fragment"
  xmlns=""
  fragbodyref="http://www.w3.org/TR/REC-xml.html#sec-xml-and-sgml"
  extref="http://www.w3.org/TR/REC-html40-971218/loose.dtd">
  <html>
    <head>
      <link rel='STYLESHEET' type='text/css' href='/StyleSheets/TR/rec.css'/>
    </head>
    <body>
      <h1>Extensible Markup Language (XML) 1.0</h1>
      <h2 ID='sec-intro'>1. Introduction</h2>
      <h3 ID='sec-origin-goals'>1.1 Origin and Goals</h3>
      <h3 ID='sec-terminology'>1.2 Terminology</h3>
      <h2 ID='sec-documents'>2. Documents</h2>
      <h3 ID='sec-well-formed'>2.1 Well-Formed XML Documents</h3>
      <h3 ID='charsets'>2.2 Characters</h3>
      <h3 ID='sec-common-syn'>2.3 Common Syntactic Constructs</h3>
      <h3 ID='syntax'>2.4 Character Data and Markup</h3>
      <h3 ID='sec-comments'>2.5 Comments</h3>
      <h3 ID='sec-pi'>2.6 Processing Instructions</h3>
      <h3 ID='sec-cdata-sect'>2.7 CDATA Sections</h3>
      <h3 ID='sec-prolog-dtd'>2.8 Prolog and Document Type Declaration</h3>
      <h3 ID='sec-rmd'>2.9 Standalone Document Declaration</h3>
      <h3 ID='sec-white-space'>2.10 White Space Handling</h3>
      <h3 ID='sec-line-ends'>2.11 End-of-Line Handling</h3>
      <h3 ID='sec-lang-tag'>2.12 Language Identification</h3>
      <h2 ID='sec-logical-struct'>3. Logical Structures</h2>
      <h3 ID='sec-starttags'>3.1 Start-Tags, End-Tags, and Empty-Element Tag</h3>
      <h3 ID='elemdecls'>3.2 Element Type Declarations</h3>
      <h4 ID='sec-element-content'>3.2.1 Element Content</h4>
      <h4 ID='sec-mixed-content'>3.2.2 Mixed Content</h4>
```

```

<h3 ID='attdecls'>3.3 Attribute-List Declarations</h3>
<h4 ID='sec-attribute-types'>3.3.1 Attribute Types</h4>
<h4 ID='sec-attr-defaults'>3.3.2 Attribute Defaults</h4>
<h4 ID='AVNormalize'>3.3.3 Attribute-Value Normalization</h4>
<h3 ID='sec-condition-sect'>3.4 Conditional Sections</h3>
<h2 ID='sec-physical-struct'>4. Physical Structures</h2>
<h3 ID='sec-references'>4.1 Character and Entity References</h3>
<h3 ID='sec-entity-decl'>4.2 Entity Declarations</h3>
<h4 ID='sec-internal-ent'>4.2.1 Internal Entities</h4>
<h4 ID='sec-external-ent'>4.2.2 External Entities</h4>
<h3 ID='TextEntities'>4.3 Parsed Entities</h3>
<h4 ID='sec-TextDecl'>4.3.1 The Text Declaration</h4>
<h4 ID='wf-entities'>4.3.2 Well-Formed Parsed Entities</h4>
<h4 ID='charencoding'>4.3.3 Character Encoding in Entities</h4>
<h3 ID='entproc'>4.4 XML Processor Treatment of Entities and Reference:
<h4 ID='not-recognized'>4.4.1 Not Recognized</h4>
<h4 ID='included'>4.4.2 Included</h4>
<h4 ID='include-if-valid'>4.4.3 Included If Validating</h4>
<h4 ID='forbidden'>4.4.4 Forbidden</h4>
<h4 ID='inliteral'>4.4.5 Included in Literal</h4>
<h4 ID='notify'>4.4.6 Notify</h4>
<h4 ID='bypass'>4.4.7 Bypassed</h4>
<h4 ID='as-PE'>4.4.8 Included as PE</h4>
<h3 ID='intern-replacement'>4.5 Construction of Internal Entity Replac
<h3 ID='sec-predefined-ent'>4.6 Predefined Entities</h3>
<h3 ID='Notations'>4.7 Notation Declarations</h3>
<h3 ID='sec-doc-entity'>4.8 Document Entity</h3>
<h2 ID='sec-conformance'>5. Conformance</h2>
<h3 ID='proc-types'>5.1 Validating and Non-Validating Processors</h3>
<h3 ID='safe-behavior'>5.2 Using XML Processors</h3>
<h2 ID='sec-notation'>6. Notation</h2>
<h3>Appendices</h3>A. <A ID='sec-bibliography'>References</A>
<h3 ID='sec-existing-stds'>A.1 Normative References</h3>
<h3 ID='null'>A.2 Other References</h3>
<h2 ID='CharClasses'>B. Character Classes</h2>
</f:fragbody/>
<h2 ID='sec-entexpand'>D. Expansion of Entity and Character References
<h2 ID='determinism'>E. Deterministic Content Models (Non-Normative)</h2>
<h2 ID='sec-guessing'>F. Autodetection of Character Encodings (Non-Norr
<h2 ID='sec-xml-wg'>G. W3C XML Working Group (Non-Normative)</h2>
</body>
</html>
</f:fcs>

```

## D Design Principles (Non-Normative)

In the design of any language, trade-offs in the solution space are necessary. To aid in making these trade-offs the follow design principles were used (the order of these principles is not necessarily significant):

1. XML fragment specifications should be usable over the internet.
2. XML fragment specifications should support the specification of context for any well-formed chunk of XML; the definition of a fragment may be broadened to allow any chunk of XML that matches XML's "content" production (production [43]). Chunks of XML that do not match XML's

“content” production (i.e., that are not well-formed entities) are specifically out of scope.

3. XML fragment specifications should be optimized to work with simpler XML fragments (such as those conforming to the simpler XML profile being developed by the XML Syntax WG), though the language should also work with any XML (“the easy stuff should be easy, and the harder stuff should be possible”); working with SGML features not included in XML (including those, such as tag omission, allowed in HTML) is not a goal.
4. XML fragment specifications should be capable of being specified both in the same storage object as the fragment body itself as well as in a separate object linked in some fashion to the fragment body.
5. XML fragment specifications should support interaction with XML browsers, editors, repositories, and other XML applications.
6. SGML features and characteristics not included in XML shall not be taken into consideration in the design of our fragment context specification solution.
7. It is specifically not a goal that XML fragment specifications be designed in consideration of non-XML HTML browsers, parsers, or other non-XML applications.
8. Since interoperability is a primary goal, there should be only one language for the fragment context specification rather than multiple “features.” However, since the goal is to provide enough information to parse the fragment, and well-formed XML may not require any extra information to allow it to be parsed, no specific set of context information should be required in all context specifications. (No implementation should choke on any valid piece of context information, but no implementation should be considered non-compliant for choosing to ignore [on the receiving end]—or not include [on the sending end]—a specific piece of context information if doing so makes sense in the particular environment.)
9. XML fragment specifications should leverage other recommendations and standards, including XML 1.0, XML Namespace, XPointer, XML Information Set, the SGML Open TR9601:1996 on Fragment Interchange, and relevant IETF work.
10. XML fragment specifications should be human-readable and reasonably

clear.

11. Terseness in XML fragment specification syntax is of minimal importance.
12. Issues involved with the possible “return” of any fragment to its original context and the determination of the possible validity of the “returned” fragment in its original context are beyond the scope of this activity.

## E Acknowledgments (Non-Normative)

The following participated in the XML Fragment WG during the authoring of this Recommendation:

Paula Angerstein, Vignette  
Tim Boland, NIST  
Charles Frankston, Microsoft  
Paul Grosso, Arbortext  
Michael Hyman, Microsoft  
Joel Nava, Adobe  
Conleth O'Connell, Vignette  
Joakim Östman, Citec  
Christina Portillo, Boeing  
Shriram Revankar, Xerox  
Daniel Veillard, W3C

## F Changes from Previous Public Working Drafts (Non-Normative)

### F.1 Changes between the March 3 and April 2 WD

Major changes to the previous public working draft are outlined below. Various other changes have also been made throughout the document.

1. Added [fragment context specification document] as a defined term.
2. Added a `fragbodyref` attribute to the `fragbody` element ([PROD: 4]) and renamed the `fragbodyref` attribute of the `fcs` element to `source1ocn`.
3. Added a production ([PROD: 7]) to allow an `fcs` to have a prolog; added a

well-formed, namespace complete FCS Constraint.

4. Wrote a new subsection of the fcs notation chapter (5.3 Semantics of a fragment context specification) describing the Semantics of a fragment context specification.
5. Wrote a new subsection of the fcs notation chapter (5.4 An fcs example) giving a complete example of a fragment context specification use (without packaging).
6. Moved the chapter on packaging to the non-normative back matter (B Packaging and interchanging fragments).
7. Did major editing of the appendix of examples (C Examples).

## F.2 Changes between the April 2 and June 19 WD

Major changes to the previous public working draft are outlined below. Various other minor changes have also been made to the document.

1. The Status section was updated.
2. References to XPointer usage were replaced with references to “URI reference [RFC 2396].”
3. Some items in the fragment context information set were moved from the “affect proper parsing” list to the “cannot affect proper parsing” list.
4. An additional note was added at the top of the Overview of the fcs to indicate what kinds of fci is necessary and sufficient for CSS use.
5. The conformance section was expanded.
6. References to related IETF RFC's were added.
7. Example C.2 was modified to use content ids.

## F.3 Changes between the June 19 WD and the CR

Changes to the previous public working draft are outlined below.

1. The Status section was updated.
2. The Decision notes and review requests were removed.
3. IDs were added on various elements to allow for more granular referencing.